

Forum:[Hack Forums](#)

Thread: [BackMeUp - Python local file backup \[Updated v1.1\]](#)

glob - August 6, 2014, 10:56 am

Just whipped this up in about an hour because I was bored. It's just a python script to back up your files to an external drive.

<http://github.com/globby/BackMeUp>

dberube - August 6, 2014, 3:24 pm

Nice little script. The code quality is top notch so I can't say anything about that.

I don't know if the script will run with python 3, but if it can you could optionally add support for the lzma compression algorithm to get the best compression ratio.

<https://docs.python.org/3/library/lzma.html#module-lzma>

CodeX - August 6, 2014, 6:01 pm

I guess this could be fairly handy for the circumstances it was made for and the code itself is quite nice, but almost half of it's functionality can be done with [one function](#) - so here's some waffle about your implementation. You can use `os.makedirs` to create any needed leading directories instead of doing it yourself, but by default the tree is walked top-down so in theory you would have always created the leading directories after the root if you see them at all. If you are just copying a tree then `shutil.copytree` may be the way to go

If you are going to use `os.path.join` then it's probably a good idea to use `os.path.split` instead of `p.split('\\')`

Playing with NTFS's alternate data streams may not be necessary e.g. NTFS->NTFS or when zipping (I guess a zipped file name can technically have a `:` in it and it's hopefully not a problem on other filesystems [as they tend to allow `:`]) You could also end up with problems quietly occurring if for some reason `"%s%s"%(FileName1,ADSName) == FileName2`

does `os.walk` actually report ADS on Windows? I doubt it would on Linux as the driver treats the streams as extended attributes but someone might have thought of this. Extendy bits: Copying additional file attributes like modification/access times may be useful - this can be done by changing `shutil.copy` to `shutil.copy2`

By default `os.walk` doesn't follow links, this might be something that you would want to do in certain circumstances

Incremental backups may be useful

Keeping only the last few backups may also be useful

If you are going to repeat backing up then it is probably a good idea to avoid exiting as soon as you can't access the backup drive - poll the drive (or otherwise get a signal of it's return) until it available again and then continue the backup. Just exiting may be perfectly fine for your usage scenario though

You can easily end up with multiple drives being assigned the same drive name which may not do what you want - IIR on Windows you can access drives by using their [GUID](#) with something like this `\\?\\{3F2504E0-4F89-41D3-9A0C-0305E82C3301}\\DIRNAME\\BASENAME`

And so that's (almost) my weekly waffling done

Spoiler ([Click to View](#))

[img]http://crescentflame.files.wordpress.com/2013/03/shamelessplug.jpg[/img]

I made something vaguely similar to backup/duplicate files a month or two ago but it uses UDisks2's dbus signals (straightforward if you are on Ubuntu) to detect when a drive is connected via it's GUID and then it checks to make sure that all of the source drives for the task are connected, if so it then copies any modified files over to the target drive and then turns off the drive when it is done to show that it can be removed. I use it to synch directories/files between my server and removable drives so I can have portable copies of all my movies/series instead of a brazillion DVDs.

I might tart it up and release it some day if there's enough interest of I have other reason to make it sufficiently portable

glob - August 6, 2014, 7:33 pm

(Aug 6, 2014 06:01 PM)CodeX Wrote: [\[CITE\]](#) I guess this could be fairly handy for the circumstances it was made for and the code itself is quite nice, but almost half of it's functionality can be done with [one function](#) - so here's some waffle about your implementation You can use os.makedirs to create any needed leading directories instead of doing it yourself, but by default the tree is walked top-down so in theory you would have always created the leading directories after the root if you see them at all. If you are just copying a tree then shutil.copytree may be the way to go

If you are going to use os.path.join then it's probably a good idea to use os.path.split instead of p.split('\\')

Playing with NTFS's alternate data streams may not be necessary e.g. NTFS->NTFS or when zipping (I guess a zipped file name can technically have a : in it and it's hopefully not a problem on other filesystems [as they tend to allow :]) You could also end up with problems quietly occurring if for some reason "%s%s"%(FileName1,ADSName) == FileName2

does os.walk actually report ADS on Windows? I doubt it would on Linux as the driver treats the streams as extended attributes but someone might have thought of this Extendy bits: Copying additional file attributes like modification/access times may be useful - this can be done by changing shutil.copy to shutil.copy2

By default os.walk doesn't follow links, this might be something that you would want to do in certain circumstances

Incremental backups may be useful

Keeping only the last few backups may also be useful

If you are going to repeat backing up then it is probably a good idea to avoid exiting as soon as you can't access the backup drive - poll the drive (or otherwise get a signal of it's return) until it available again and then continue the backup. Just exiting may be perfectly fine for your usage scenario though

You can easily end up with multiple drives being assigned the same drive name which may not do what you want - IIR on Windows you can access drives by using their [GUID](#) with something like this \\?\\{3F2504E0-4F89-41D3-9A0C-0305E82C3301}\\DIRNAME\\BASENAME

And so that's (almost) my weekly waffling done

Spoiler ([Click to View](#))

[img]http://crescentflame.files.wordpress.com/2013/03/shamelessplug.jpg[/img]

I made something vaguely similar to backup/duplicate files a month or two ago but it uses UDisks2's dbus signals (straightforward if you are on Ubuntu) to detect when a drive is connected via it's GUID and then it checks to make sure that all of the source drives for the task are connected, if so it then copies any modified files over to the target drive and then turns off the drive when it is done to show that it can be removed. I use it to synch directories/files between my server and removable drives so I can have portable copies of all my movies/series instead of a brazillion DVDs.

I might tart it up and release it some day if there's enough interest of I have other reason to make it sufficiently portable

Thank you for your extended feedback, I'll certainly update the script accordingly (I wasn't actually aware of shutil.copytree before this point ?)

Edit: Made some changes based on your suggestions

(Aug 6, 2014 03:24 PM)dberube Wrote: [\[CITE\]](#) Nice little script. The code quality is top notch so I can't say anything about that.

I don't know if the script will run with python 3, but if it can you could optionally add support for the lzma compression algorithm to get the best compression ratio.

<https://docs.python.org/3/library/lzma.html#module-lzma>

It would probably work if I ran 2to3 on it (I'm not extremely familiar with python 3, but it seems like print would be the only thing that would fuck it up). I'll keep that in mind, thanks :)

CodeX - August 7, 2014, 4:49 am

Welcome :3

I guess that this would work in Python3 if the print parameters were enclosed in brackets - in this script it wouldn't be a problem as you only ever pass one parameter to print. If you had multiple parameters to print then you would end up passing a tuple to the print statement which would mess things up but with a single parameter it just wraps the parameter in brackets and so works fine. You can also do [import from __future__](#) to treat print as a function instead of a statement but given your print style it wouldn't really be needed.

I went off on one about handling [ADS](#) last night when your `.replace(':')` was (in hindsight) probably to sanitise the drive letter, you might find that using paths relative to the directory you are backing up (when zipping) would be more appropriate which could be done with `os.path.relpath`, you can also get the drive by using [os.path.splitdrive](#).

If you feel like playing with some (possibly) new bits for this then you could [get a message](#) from the Windows system when a USB device is connected. You'd probably end up using [ctypes](#) to call the relevant OS functions which you may or may not find fun but it can be nice to know how to work with native functions if you are into tinkering.

Black?jack - August 7, 2014, 7:24 am

(Aug 6, 2014 10:56 AM)glob Wrote: [\[CITE\]](#) Just whipped this up in about an hour because I was bored. It's just a python script to back up your files to an external drive.

<http://github.com/globby/BackMeUp>

I love you I love you I love you so much.

I thought I was the only one that did that!

glob - August 7, 2014, 11:33 am

(Aug 7, 2014 07:24 AM)Black?jack Wrote: [\[CITE\]](#) I love you I love you I love you so much.

I thought I was the only one that did that!

<http://legacy.python.org/dev/peps/pep-0008/#imports>

Cisteriaso - August 7, 2014, 11:55 am

Thanks for sharing it. I might take a look at it later.

Black?jack - August 7, 2014, 4:30 pm

(Aug 7, 2014 11:33 AM)glob Wrote: [\[CITE\] http://legacy.python.org/dev/peps/pep-0008/#imports](http://legacy.python.org/dev/peps/pep-0008/#imports)
I mean the organizing in order of length of the module name :P

EPIXIA???? - August 7, 2014, 6:39 pm

(Aug 7, 2014 04:30 PM)Black?jack Wrote: [\[CITE\]](#) I mean the organizing in order of length of the module name :P I do that too, it has to look pretty :3

glob - August 7, 2014, 8:39 pm

(Aug 7, 2014 04:30 PM)Black?jack Wrote: [\[CITE\]](#) I mean the organizing in order of length of the module name :P

Oh, yeah I do that without even thinking XD